<u>Amendments to the Claims:</u>

This listing of claims will replace all prior versions, and listings, of claims in the application:

<u>Listing of Claims:</u>

1.   (Canceled)

2.   (Currently Amended) A <u>computer implemented</u> method for selecting a class loader to load a plug-in class within a class loader heirarchy, the method comprising the steps of:

generating a class loader hierarchy comprising a plurality of class loaders that includes two or more application class loaders, one for each of two or more application classes, wherein each of said application class loaders can selectively load its application class, or delegate the loading of its application class to another class loader of said class loader hierarchy;

providing a different plug-in class loader for each class loader [[in the]] <u>of said</u> class loader hierarchy, wherein <u>each plug-in class loader is associated with only a single class loader of said hierarchy, and</u> each plug-in class loader delegates to its ~~corresponding~~ <u>associated</u> class loader;

identifying the class loader of said hierarchy that is used to load a specified one of said two or more application classes; and

using the plug-in class loader that is ~~provided for~~ <u>associated with</u> and delegates to said identified class loader to load a <u>given</u> plug-in class that is associated with said specified application class.

3.   (Currently Amended) The method of claim 2, wherein ~~the plurality of class loaders includes a boot class loader~~ <u>an application file is associated with said plug-in classes, and said plug-in class loader that is associated with said identified class loader is enabled to locate said given plug-in class by specifying a single configuration value in said application file.</u>

4.   (Currently Amended) The method of claim 3, wherein the plurality of class loaders includes an extension class loader, wherein the extension class loader delegates to [[the]] <u>a</u> boot class loader.

5.   (Original) The method of claim 4, wherein the plurality of class loaders includes a system class loader, wherein the system class loader delegates to the extension class loader.

6.   (Previously Presented) The method of claim 5, wherein the two or more application class loaders delegate to the system class loader.

7.   (Previously Presented) The method of claim 2, further comprising:

identifying said plug-in class loader that is provided for and delegates to said identified class loader.

8.   (Original) The method of claim 2, further comprising:

responsive to a first application class loading a first plug-in class, identifying a target class loader within the class loader hierarchy that loaded a target class;

identifying a plug-in class loader that is provided for and delegates to the target class loader; and

loading the first plug-in class using the plug-in class loader.

9.   (Original) The method of claim 8, wherein the step of identifying a target class loader within the class loader hierarchy that loaded a target class includes using a class loader that loaded the application class to look up the target class.

10.   (Currently Amended) The method of claim 8, further comprising 2, wherein:

a particular one of said application classes is loaded by a particular class loader of said class loader hierarchy, and a first plug-in class associated with a first application class and a second plug-in class associated with a second application class are each specified to use the class loader of said particular application; and

said first plug-in class and said second plug-in class are both loaded by the plug-in class loader that is associated with and delegates to said particular class loader.

responsive to a second application class loading a second plug-in class, identifying the target class loader within the class loader hierarchy that loaded the target class;

identifying the plug-in class loader that is provided for and delegates to the target class loader; and

loading the second plug-in class using the plug-in class loader.

11.   (Currently Amended) The method of claim 10, wherein [[the]] said first plug-in class and [[the]] said second plug-in class are able to share data.

12.   (Currently Amended) An apparatus for selecting a class loader to load a plug-in class within a class loader hierarchy, the apparatus comprising:

means for generating a class loader hierarchy comprising a plurality of class loaders that includes two or more application class loaders, one fore each of two or more application classes, wherein each of said application class loaders can selectively load its application class, or delegate the loading of its application class to another class loader of said class loader hierarchy;

means for providing a different plug-in class loader for each class loader [[in the]] of said class loader hierarchy, wherein each plug-in class loader is associated with only a single class loader of said hierarchy, and each plug-in class loader delegates to its ~~corresponding~~ associated class loader;

means for identifying the class loader of said hierarchy that is used to load a specified one of said two or more application classes; and

means for using the plug-in class loader that is ~~provided for~~ associated with and delegates to said identified class loader to load a plug-in class that is associated with said specified application class

13.   (Original) The apparatus of claim 12, wherein the plurality of class loaders includes a boot class loader.

14.   (Original) The apparatus of claim 13, wherein the plurality of class loaders includes an extension class loader, wherein the extension class loader delegates to the boot class loader.

15.   (Original) The apparatus of claim 14, wherein the plurality of class loaders includes a system class loader, wherein the system class loader delegates to the extension class loader.

16.   (Previously Presented) The apparatus of claim 15, wherein the two or more application class loaders delegate to the system class loader.

17.   (Previously Presented) The apparatus of claim 12, further comprising:

means for identifying said plug-in class loader that is provided for and delegates to said identified class loader.

18.   (Original) The apparatus of claim 12, further comprising:

responsive to a first application class loading a first plug-in class, identifying a target class loader within the class loader hierarchy that loaded a target class;

identifying a plug-in class loader that is provided for and delegates to the target class loader; and

loading the first plug-in class using the plug-in class loader.

19.   (Original) The apparatus of claim 18, wherein the means for identifying a target class loader within the class loader hierarchy that loaded a target class includes means for using a class loader that loaded the application class to look up the target class.

20.   (Original) The apparatus of claim 18, further comprising:

means, responsive to a second application class loading a second plug-in class, for identifying the target class loader within the class loader hierarchy that loaded the target class;

means for identifying the plug-in class loader that is provided for and delegates to the target class loader; and

means for loading the second plug-in class using the plug-in class loader.

21.   (Original) The apparatus of claim 20, wherein the first plug-in class and the second plug-in class share data.

22.   (Currently Amended) A computer program product, in a computer readable medium, for selecting a class loader to load a plug-in class within a class loader hierarchy, the computer program product comprising:

first instructions for generating a class loader hierarchy comprising a plurality of class loaders that includes two or more application class loaders, one for each of two or more application classes, wherein each of said application class loaders can selectively load its application class, or delegate the loading of its application class to another class loader of said class loader hierarchy;

second instructions for providing a different plug-in class loader for each class loader [[in the]] of said class loader hierarchy, wherein each plug-in class loader is associated with only a single class loader of said hierarchy, and each plug-in class loader delegates to its corresponding associated class loader;

third instructions for identifying the class loader of said hierarchy that is used to load a specified one of said two or more application classes; and

fourth instructions for using the plug-in class loader that is provided for associated with and delegates to said identified class loader to load a plug-in class that is associated with said specified application class.

Amendments to the Drawings:

Figures 1-3 have each been amended herewith to include the legend --PRIOR ART--.

Attachments: 2 Replacement Sheets

                      2 Annotated Sheets Showing Changes